In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy.

Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments

Edoardo Gaetani[1], Leonardo Aniello[1], Roberto Baldoni[1], Federico Lombardi[1], Andrea Margheri[2], and Vladimiro Sassone[2]

[1] Research Center of Cyber Intelligence and Information Security, La Sapienza University of Rome
{surname}@dis.uniroma1.it
[2] University of Southampton
{a.margheri;vsassone}@soton.ac.uk

Abstract

Data is nowadays an invaluable resource, indeed it guides all business decisions in most of the computer-aided human activities. Threats to *data integrity* are thus of paramount relevance, as tampering with data may maliciously affect crucial business decisions. This issue is especially true in *cloud computing environments*, where data owners cannot control fundamental data aspects, like the physical storage of data and the control of its accesses. *Blockchain* has recently emerged as a fascinating technology which, among others, provides compelling properties about data integrity. Using the blockchain to face data integrity threats seems to be a natural choice, but its current limitations of low throughput, high latency, and weak stability hinder the practical feasibility of any blockchain-based solutions.
In this paper, by focusing on a case study from the European SUNFISH project, which concerns the design of a secure by-design cloud federation platform for the public sector, we precisely delineate the actual data integrity needs of cloud computing environments and the research questions to be tackled to adopt blockchain-based databases. First, we detail the open research questions and the difficulties inherent in addressing them. Then, we outline a preliminary design of an effective blockchain-based database for cloud computing environments.

1 Introduction

Data is nowadays a key asset. It is strategic to drive any business decision in countless different fields, ranging from finance and insurance, to health, education and public administration. As computer-aided human activities are relying more and more on data, trusting data has thus become crucial. At the same time, the critical role of data has made it a very appealing target for cyber-attacks, which aim at undermining the fundamental CIA properties (Confidentiality, Integrity, Availability) that data should exhibit in order to be trusted.

Cyber-attacks against CIA properties cause different impairments on data trust according to the undermined property. Specifically, sabotaging availability prevents data to be retrieved only for temporary period of time, but operations can be resumed as soon as data are accessible again. Compromising confidentiality discloses instead private data and cannot be reverted, but original data are still available and usable, at least to the extent allowed by the inflicted damage (i.e., an organisation victim of data leakage may have to face economic consequences). Instead, tampering with data integrity is a highly damaging attack that always paves critical issues to data trust. Indeed, tampering with data can go undetected and drive operations maliciously, by deleting specific entries (i.e., to remove inconvenient traces) or by altering particular sections of data (i.e., to change data consumers' behaviour). In 2015, Kaspersky Lab found out a massive cyber-attack targeting over than 100 financial institutes worldwide that siphoned off money from account balances for an estimated value of around $1 billion[1]. Differently from confidentiality

[1] https://securityintelligence.com/sabotage-the-latest-threat-to-the-financialbanking-industry/

and availability, once integrity is compromised there is no way to restore the original data, it is lost forever. Therefore, as integrity attacks are subtle to be detected and really effective, in this paper we focus on data integrity rather than confidentiality or availability.

Data integrity issues are exacerbated in cloud computing environments, as data owners hardly control where their data are stored, who can actually access them, and in which way. Nevertheless, more and more private and public organisations are outsourcing their data, because "it relieves the burden of maintenance cost as well as the overhead of storing data locally" [10]. Therefore, ensuring data integrity properties in cloud computing environments has become an urgent need to address.

Data integrity is commonly assured by using, on one hand, cryptographic tools (i.e., digests, asymmetric keys) and, on the other hand, appropriate data replication strategies. The cryptographic tools are indeed used to sign single pieces of data, so that any forging attack can be promptly detected via cryptographic signature validations. Indeed, an attack to be effective would require the violation of the secret keys, thus to update data signatures and circumvent the cryptographic integrity checks. These attacks are challenging to carry out, but once realised they are practically undetectable. Therefore, it is highly advocated to exploit appropriate data replication strategies to ensure anyhow data integrity.

Replicating and distributing data over a set of nodes critically hamper the violation of data integrity: an attacker should compromise, without being detected, all the replicated data. This replication approach is widely adopted in practice, like, e.g., in the context of cloud computing environments, where there is abundance of distributed storage resources. However, although replication surely increases the burden for a successful attack in a cloud setting, cloud providers themselves can collude with attackers for easily violating data integrity. To inhibit these collusion attacks and to avoid blind trust on the integrity guarantees claimed by cloud providers, we advocate an innovative exploitation of the *blockchain* technology to design and implement a distributed, secure *blockchain-based database* for cloud computing environments.

Intuitively, a blockchain can be seen as a replicated database distributed among thousands of nodes belonging to diverse parties. In its first conception it has been used as public ledger for Bitcoin transactions [8]. Recently, it has gained a great momentum for the fascinating properties it guarantees (e.g., distributed consensus, persistent and non-repudiable data) Among others, the practical inability to alter an information that has been stored in a blockchain for "some time" is the key focal point for data integrity. However, the actual length of such a time cannot be fixed a priori and it becomes an insurmountable obstacle to effectively exploit blockchain. For instance, a Bitcoin transaction is deemed tamper-proof about one hour after its insertion in the blockchain; clearly an unfeasible period of time for, e.g., cloud computing applications.

Despite the time-related obstacle, the intrinsic replication and distribution features of blockchains prompt their wide adoption in cloud settings. In this paper, we first shed light, by introducing a few open research questions, on the issues of employing blockchain "as-is". Then, we present practical research directions that lead the way to effective blockchain-based databases for cloud computing environments. On the one hand, we elaborate on the issues of time-dependent integrity, lack of performance and absence of stability. On the other hand, we propose an innovative blockchain-based database that permits balancing strong integrity guarantees with appropriate performance and stability properties.

Structure of the paper. We first focus in Section 2 on a specific class of scenarios related to SUNFISH, a European project about secure-by-design cloud federation, and to the data threats its case studies prompt. In Section 3 we describe the blockchain technology, its data integrity properties and current limitations, while we outline in Section 4 the research questions to address to realise effective blockchain-based databases. In Section 5 we present our solution tackling

such questions and its application to the SUNFISH cloud federation case study. Finally, after a brief discussion on related work in Section 6, we draw conclusions in Section 7.

2 Case Study: the European SUNFISH Project

Nowadays, an urgent need of public and private companies is to prompt and support interoperability and cooperation among their already deployed (private) cloud systems (see, e.g., the ENISA report in [4]). Indeed, it is advocated that different cloud systems federate themselves into goal-oriented federations. Besides the multiple technical issues to address, the creation and management of cloud federations have to face daunting security issues, mainly related to the non-disclosure of sensitive data and the enforcement of integrity guarantees. To overcome these security difficulties, the EU SUNFISH project aims at proposing a distributed, democratic cloud federation platform that will ensure by-design the security of the managed data.

The SUNFISH proposal is *Federation-as-a-Service* (FaaS) [9], a new and innovative service that enables the secure creation and management of cloud data and services. FaaS features advanced data security services and innovative design principles leading to a distributed and democratic cloud federation governance. For the sake of presentation, we do not comment on the data security services (further details are reported, e.g., in [12, 13]), while we extensively address the role of data integrity in federation governance.

The intrinsic goal of cloud federations is sharing services among members by creating regulated, secured inter-cloud interactions. The rules governing these interactions, hence the service usage, are defined in specific contracts. For instance, a member providing a service may require that only specific consumers can use it and that the service outputs have to be masked for privacy reasons. Due to the high sensitivity of the data managed by cloud federations (e.g., personal and medical data in case of the public sector), FaaS must provide high assurances about the compliance of the member contracts. Indeed, besides the runtime enforcing of the contracts, FaaS has to guarantee the integrity of contracts, namely that they cannot be tampered with and that all involved members must be aware of their existence. Additionally, to ensure non-repudiable evidences of contract enforcement, all the inter-cloud interactions have to be monitored and the logs stored with strong integrity guarantees.

Most of all, to foster a wide adoption of cloud federations, FaaS advocates the absence of a centralised governance. As a matter of fact, among federation members there cannot be designed a leader (i.e., there is no *primus inter pares*), rather federation members form a network of peers. To this aim, FaaS seeks to establish a decentralised, democratic federation governance, hence it must rely on an opportunely defined, distributed database ensuring strong integrity guarantees. The novel design solution for FaaS advocated by the SUNFISH project is based on the exploitation of a blockchain. To properly address the feasibility of such a solution, significant threats to data integrity have to be identified.

2.1 Threats to Data Integrity

The threats to the data integrity in the context of cloud federation can be multiple and variegated. Our focus is on the database storing the governance data of a federation, hence on data whose corruption critically affects the whole federation and its security. The threats we consider span from malicious alterations of data, to data updates without all the involved members informed. More specifically, we can enumerate the following threats:

T1 An attacker violates the integrity of the data by directly altering (part of) the database.

T2 A federation member updates the database without informing the other members.

T3 Multiple federation members collude to maliciously altering (part of) the database.

Threat **T1** is straightforward, while Threat **T2** is due to the democratic nature of a SUNFISH federation. For instance, adding to the database a log entry about a fake inter-cloud interaction between members A and B, hence without the A and B being informed, is a clear integrity violation. Therefore, the process of adding data to the database should rely on opportunely devised consensus schemas. However, as pointed out in Threat **T3**, even consensus schemas can be attacked: federation members can collude together to alter the database integrity. For instance, given a member A providing to the federation a service s, the other members can collude to compromise s by, e.g., storing false information on the service (i.e., altering the contracts regulating the provisioning of s, or removing log entries about inappropriate uses of s) to obtain advantages and causing the detriment of A.

3 Blockchain: Data Integrity, Performance, Stability

The blockchain is a quite novel technology that has appeared on the market in the recent years, firstly used as public ledger for the Bitcoin cryptocurrency [8]. It mainly consists of consecutive chained blocks containing records, that are replicated on the nodes of a p2p network. These records witness transactions occurred between pseudonyms. Transactions may feature a cryptocurrency like, e.g., the Bitcoin, or other kinds of assets. The collection of transactions and their enclosing in chain blocks is carried out in a decentralised fashion by distinguished nodes of the network, i.e. *miners*. Miners apply opportune block construction methods, i.e., the *mining process*, to achieve consensus among all the miners on newly generated blocks. Bitcoin is an example of *permissionless* blockchain, i.e., there is no restriction for a node to become a miner. If instead there is an authentication and authorization layer for miners, then the blockchain is *permissioned*.

The original mining process, still used for Bitcon and Ethereum [14] blockchain, is based on the *proof of work* (PoW). It consists in a computational intensive hashing task that is regulated according to the so-called *blockchain difficulty* that regulates the average time spent by miners to accomplish such a task and create a new block. Once a miner achieves the creation of a new block, it broadcasts that block to all the other miners. They consider such a block as the latest of the chain and start mining new blocks to be appended. For the sake of simplicity, we can say that once a miner has created a new block, it becomes part of the chain (if multiple miners concurrently add a block, a transient fork is created which is usually quickly resolved because by design miners always consider the longest chain).

PoW-based blockchains enjoy many fascinating properties related to *data integrity*, which follow from the mining process and from the full replication of the blockchain on a large number of nodes. Indeed, when a block is part of the chain, all miners have agreed on its contents, hence it is practically non-repudiable and persistent (unless an attacker has the majority of miners' hash power that are able to create a fork of the chain). Assuming a majority of hash power controlled by honest miners, the probability of a fork of depth n is $\mathcal{O}(2^{-n})$ [2]. This gives users high confidence that simply waiting for a small number of nodes to be added (6 blocks in Bitcoin) will ensure their transactions are permanently included with high confidence.

However, PoW-based blockchains have a main drawback: *performance*. This lack of performance is mainly due to the broadcasting latency of blocks on the network and the time-intensive task of PoW. As a matter of fact, each transaction stored on a blockchain has a high confirmation latency, which causes an extremely low transaction throughput. In Bitcoin, the average

latency is 10 minutes, and the throughput is about 7 transactions per second [2].

Another relevant concern related to the use of the blockchain regards its *stability*. Although, e.g., the Bitcoin's blockchain has worked quite well so far, there is no universally accepted academic work explaining either why this has happened, or whether it will continue in the future, or how long it will [2]. The stability properties of the PoW-based consensus protocol are still being debated, and current "literature does not even provide adequate tools to assess under which economic and social assumptions Bitcoin itself will remain stable" [2]. In general, PoW-based blockchains using incentive mechanisms based on cryptocurrencies are heavily subjected to market fluctuations, which casts a shadow on the blockchain effectiveness on the long term.

4 Open Research Questions

In the context of cloud computing environments, the blockchain could be exploited to realise a database ensuring strong integrity guarantees. In particular, the blockchain could be used to store the logs of database operations, thus to avoid the data threats presented in Sec. 2.1. However, current blockchains cannot be employed "as-is" due to various deficiencies. In the following, we address the main issues related to data integrity, performance limitations, and blockchain stability.

How to Measure Data Integrity? Once data has been included in a block, if we assume a majority of honest miners, we can be highly confident that the chances of data alteration decrease exponentially over time (see Sec. 3). Data integrity is indeed strictly related to these chances. The more unlikely it is that data can be tampered with, the stronger the integrity guarantees we can claim. However, being dependent on time introduces critical aspects to address for ensuring data integrity: there is hardly information on the effective integrity guarantees once a transaction has just been sent to the blockchain. These observations suggest that data integrity on blockchain cannot be simply seen as a binary property, which either holds or does not, but it should be intended as a more complex, quantitative concept. This amounts to take multiple factors into account, including the time and parties' awareness. The described issues thus lead us to formulate this research question:

Q1 *How can we quantitatively characterise data integrity guarantees, in order to enable comparison among different blockchain-based database solutions?*

Reasonable approaches to answer to this question should be based on the the effort an attacker would spend to compromise data integrity without being detected.

How to Improve Performance? The performances currently achievable with PoW-based blockchains are really poor as compared with classical database technologies. The experimented latency and throughput are almost incompatible with the requirements of the considered cloud scenarios. In this sense, a challenging and fundamental research problem regards the investigation of novel blockchain designs aimed at delivering performances aligned to today's requirements, while keeping the needed integrity guarantees.

Q2 *How can we design a blockchain-based database with better performances compared to a PoW-based blockchain "as-is", and with comparable data integrity guarantees?*

The resulting designs should be flexible enough to enable variable tradeoff between performance and integrity, thus to choose the setting that better fits the requirements.

How to Enhance Stability? Current PoW-based permissionless blockchains rely on a market-dependent cryptocurrency that may make the storing of data highly expensive and too dependent on market variations, i.e. it cannot ensure stability. Likewise poor performance cut out

many possible practical applications, unsatisfactory stability assurances can severely restrict the applicability of blockchain to database. Enhancing the stability of PoW-based permissionless blockchains, e.g. Bitcoin's, amounts to alter the currency incentives underlying the mining process. Due to the large economic interests and speculation behind cryptocurrencies, such an amendment is practically infeasible. A more viable path is exploiting permissioned blockchains, where incentives do not depend on cryptocurrencies. The described path corresponds to answering the following research question:

Q3 *How can we setup a permissioned blockchain having stronger stability compared to existing PoW-based blockchains, while preserving required guarantees on data integrity?*

The resulting blockchain will guarantee a stable support for the development of distributed databased, as it is needed, e.g., in cloud computing environments.

5 Towards an Effective Blockchain-based Database

In this section we tackle the research questions just raised and we outline our proposal for a more effective blockchain-based database. This proposal is thus intended to be part of the SUNFISH project for ensuring high data integrity guarantees (hence, to deal with the data threats presented in Sec. 2.1) and, at the same time, for being compliant with the performance and stability requirements needed in a cloud computing environment.

Our blockchain-based database aims at providing a replicated database whose integrity is testified via adequate evidences stored on a innovative designed blockchain system. Namely, we devise a two-layer blockchain that, via the first-layer, ensures adequate performance and, via a principled exploitation of the second-layer, ensures strong integrity guarantees. More specifically, the first-layer employs a lightweight distributed consensus protocol that assures low latency and high throughput. This layer aims at quickly and reliably storing evidences of every operations carried out on a distributed database. However, this layer provides weak data integrity guarantees due to the lack of PoW. Thus, the second-layer is designed as a PoW-based blockchain that stores evidences of (a part of) the database operations logged by the first-layer. These evidences are stored with strong data integrity guarantees but with poor performance. Indeed, the principled interaction between the two layers permits obtaining an overall performance improvement and effective assurances on data integrity.

In the following, we first present our proposal (Sec. 5.1), then we comment on the research questions (Sec. 5.2) and finally show how the proposal deals with the identified threats (Sec. 5.3).

5.1 A Proposal for an Effective Blockchain-based Database

In this section we introduce our proposal for an effective blockchain-based database. Blockchain is appropriately exploited to ensure the integrity of distributed replicas of a database, i.e. to store persistent evidences of the database operations that cannot be repudiated. The use of blockchain ensures not only integrity guarantees, but also fully distributed control of the database data. This intrinsic characteristic makes our proposed database feasible to be used in the context of FaaS federations. Figure 1 graphically depicts the proposed blockchain-based database distributed on three clouds member of a federation.

The member clouds operating on the database issue operations through the *Database Interface*. The operations are first logged via appropriate evidences by the the first-layer blockchain, then they are executed on the distributed *DB replica*s. More specifically, the first-layer blockchain is permissioned, and features one miner on each member cloud. The miners, by

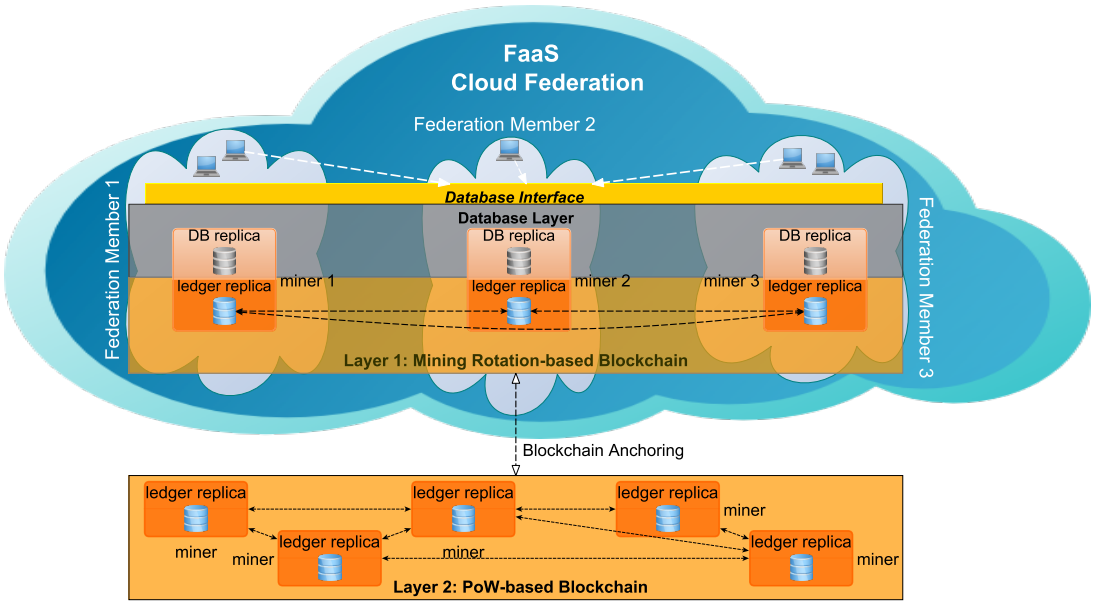

Figure 1: A blockchain-based database proposal for a Cloud Federation

relying on a public/private key pair to sign messages, achieve consensus by means of the so-called *mining rotation* consensus mechanism. Namely, it divides the time into rounds and, for each round, elects a miner as a leader. The leader is then in charge of receiving new operations, signing them with its private key, and broadcasting them to the other miners. Once all miners have signed the operations, they can become part of the blockchain: all the miners add these operations to their local *ledger*, and apply them to their local replica.

The interaction with the second-layer PoW-based blockchain is realised via a *blockchain anchoring* technique. The anchoring technique is a timed operation that permits linking a specific (part of) the first-layer blockchain with (a block of) the second-layer blockchain. In particular, at certain intervals of time, a *witness transaction* containing the hash of the first-layer blockchain up to the current operation is sent to the second-layer blockchain and, consequently, stored as immutable, irreversible transaction. These hashes act as forensics evidence for proving and validating the integrity of the data stored in the first-layer blockchain.

5.2 Preliminary Answers to the Research Questions

In the following, by referring to our proposal for blockchain-based databases, we introduce our preliminary answers to the research questions previously reported.

Measuring Data Integrity. Our answer to Question **Q1** is to measure the integrity as the effort required for an attacker to change data in the blockchain without being noticed. Quantifying this measure highly depends on the nature of the considered blockchain, but in general a desiderata is that the longer data are stored in a blockchain, the greater the effort an attacker should pay to break data integrity.

In our two-layer blockchain, the integrity measure on the evidences of a database operation is crucially affected by which of the chain contains the evidences. Indeed, if the evidences are only on the first-layer, the effort required for an attacker corresponds to compromise all the

replicas of the first-layer. However, as soon as the hash of the corresponding evidences has been stored in the second-layer, an attacker should also subvert the integrity of the PoW, thus the data integrity measure would be higher. As a matter of fact, on PoW-based blockchains like Bitcoin's, the attacker effort is close to infinite [6], i.e. it is an infeasible attack.

Improving Performance. The lack of performance of current blockchain-based systems, i.e. Question **Q2**, is due to PoW. To provide better performance to blockchain users, our proposal offers to clients a blockchain based on lightweight consensus algorithm and leverages on the power of PoW only in the background, i.e. the second-layer. Therefore, from the point of view of a client of the blockchain-based database, an operation on the database is completed as soon as it is elaborated by the first-layer blockchain.

Enhancing Stability. Permissionless blockchains like Bitcoin's and Ethereum's are natural candidates for the second-layer blockchain. To enhance their stability, i.e. Question **Q3**, we advocate the use of a blockchain that does not feature a market-dependent cryptocurrency and mining incentive mechanisms. Broadly speaking, the stability needs for blockchain highly depend on the application context. For example, in the case of the SUNFISH case studies, which address the European public sector, the need of stability is of paramount importance. In this context, we could hence envision a European permissioned PoW-based blockchain that will offer a common, stable underlying support for all European administrations.

5.3 Addressing Threats to Data Integrity

With reference to the threats outlined in Sec. 2.1, in this section we explain how the solution we propose successfully addresses them.

The effort required for an attacker to tamper with stored data (Threat **T1**) has been previously discussed in this section. At the very least, all the miners of the first-layer blockchain should be compromised, e.g. stealing their private keys. In the setting of FaaS, this would requires attacking multiple distributed cloud providers simultaneously. Even if this unlikely situation occurs, the anchoring with the second-layer blockchain ensures that only the latest set of operations on the database can be subverted; all the others are testified by immutable, irreversible evidences.

The consensus algorithm featured by the first-layer blockchain takes by design into account all the miners, hence the member clouds. Therefore, there cannot be any database operation completed without all the members being aware of it (Threat **T2**).

Collusion attacks (Threat **T3**) are instead equivalent to compromising first-layer blockchain miners. All the private keys of these miners are required to sign the messages needed to complete a database operation, thus even in the case of a single honest member attacked by a coalition formed by all the others, such honest member could successfully react. Namely, it could prevent a malicious database operation to complete by not sending its message within the consensus protocol. If the coalition attack was instead aimed to alter an information already stored in the first-layer blockchain, this means that such information has been previously agreed on by all the members, thus an honest member could then prove it owns the intact version by showing the messages previously signed and sent by the other members (when the consensus on that information was firstly achieved).

6 Related Work

Data Integrity is a well known problem in computing systems, especially for cloud environments, where users outsource their data. The task of checking data integrity for a user having relatively

poor computing devices might be very heavy due to huge amount of data to download. Ateniese et al. [1] provide one of the first model that enables a client to verify the integrity of her outsourced data on a single server without retrieving them. For a cloud environments, Remote Data Auditing (RDA) is a solution to enable auditability of outsourced data through a trusted third-party, which alleviates the computation burden on the user. A number of RDA techniques have been proposed to improve both security and efficiency, as Sookhak et al. reviewed and classified in their survey [11]. All these works rely on the assumption that the third-party is trusted. If the latter acts instead maliciously, they can no longer ensure integrity. By taking advantage of the PoW-based blockchain immutability feature,our solution is able to ensure data integrity also in a trust-less environment.

The first-layer blockchain we use to improve performance is inspired by Bitcoin-NG [5], a Bitcoin protocol modified to improve performances. To this aim they sacrifice some security guarantees, indeed data integrity is ensured only under the assumption of a majority of honest miners. Other blockchain-based databases have been proposed in literature. A similar work to ours is BigchainDB [7], a NoSQL-like storage on top of a blockchain with a built-in consensus approach. Similarly to BitCoin-NG, their main goal is to improve performances sacrificing some security guarantees. Indeed, in case of a majority of malicious miners, they can no longer ensure data integrity, similarly to Bitcoin-NG. Our solution, contrarily from both Bitcoin-NG and BigchainDB, can ensure data integrity even in case of a majority of malicious miners. Indeed, we have shown in Sec. 5.3 that we can guarantee integrity when all the miners but one are malicious and collude among themselves, therefore we can still assure integrity when overall the attacker is weaker, i.e., when a majority of miners (thus a lower number, assuming at least three miners) are malicious (thus, maybe they don't collude among themselves at all).

A remarkable work aimed at improving performance while providing security guarantees is RSCoin [3], i.e. a cryptocurrency framework that introduces a centralisation degree. A central bank maintains complete control over the monetary supply, but relies on *mintettes*, i.e. a distributed set of authorities used to prevent double-spending. They show how their solution, based on a proper consensus algorithm, allows to improve performances and ensure integrity. However, compared to our solution, their work has two main limitations: (i) a centralisation degree and (ii) integrity guarantees reached only with a majority of honest mintettes.

7 Conclusions

In this paper we identified the requirements and research questions to be addressed to realise a blockchain-based database for cloud computing environments, grounding on real needs arisen in the European project SUNFISH. Our main contribution is the proposal of a high-level solution which answers these questions, and lies the foundations for the design of a blockchain-based database able to provide the desired guarantees on data integrity, performance, and stability.

This work paves the way to appealing future works. The direction we propose can be further investigated by realising a working prototype to validate the effectiveness of our solution, in terms of achievable latency and throughput. Furthermore, a more thorough and formal examination of the tradeoff between performance and data integrity guarantees is required to prove the efficacy of our design against the identified threats. It is worth finally noticing that our blockchain-based database proposal, in this preliminary design, has been designed upon a total consensus mechanism. The approach surely permits to achieve integrity among the distributed replicas and to simplify the thread addressing. However, its availability can be critically affected by violating only a single miner. On the path of deploying such a database, we are designing an appropriate fault-tolerant consensus algorithm that permits combination

of integrity with also availability. Finally, researching on the feasibility of realising more stable blockchains is fundamental to enable their wide adoption as reliable storage infrastructures, e.g. in the context of cloud computing environments.

Acknowledgment

This work has been supported by the European Commissions H2020 Programme under the SUNFISH project, grant N.644666.

References

[1] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 598–609. ACM, 2007.

[2] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121. IEEE, 2015.

[3] George Danezis and Sarah Meiklejohn. Centrally Banked Cryptocurrencies. In *23rd Annual Network and Distributed System Security Symposium, NDSS*, 2016.

[4] ENISA. Security Framework for Governmental Clouds, 2015. Available at https://www.enisa.europa.eu/activities/Resilience-and-CIIP/cloud-computing/governmental-cloud-security/security-framework-for-govenmental-clouds.

[5] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-NG: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59, 2016.

[6] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. *The Bitcoin Backbone Protocol: Analysis and Applications*, pages 281–310. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[7] Trent McConaghy, Rodolphe Marques, Andreas Müller, Dimitri De Jonghe, Troy McConaghy, Greg McMullen, Ryan Henderson, Sylvain Bellemare, and Alberto Granzotto. BigchainDB: A Scalable Blockchain Database (DRAFT). 2016.

[8] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. Available at https://bitcoin.org/bitcoin.pdf.

[9] Francesco Paolo Schiavo, Vladimiro Sassone, Luca Nicoletti, and Andrea Margheri. FaaS: Federation-as-a-Service, 2016. Technical Report. Available at https://arxiv.org/abs/1612.03937.

[10] Mehdi Sookhak, Abdullah Gani, Hamid Talebian, Adnan Akhunzada, Samee U. Khan, Rajkumar Buyya, and Albert Y. Zomaya. Remote Data Auditing in Cloud Computing Environments: A Survey, Taxonomy, and Open Issues. *ACM Comput. Surv.*, 47(4):65:1–65:34, May 2015.

[11] Mehdi Sookhak, Hamid Talebian, Ejaz Ahmed, Abdullah Gani, and Muhammad Khurram Khan. A review on remote data auditing in single cloud server: Taxonomy and open issues. *Journal of Network and Computer Applications*, 43:121–141, 2014.

[12] Bojan Suzic, Bernd Prünster, Dominik Ziegler, Alexander Marsalek, and Andreas Reiter. Balancing Utility and Security: Securing Cloud Federations of Public Entities. In *C&TC*, volume 10033 of *LNCS*, pages 943–961. Springer, 2016.

[13] Mor Weiss, Boris Rozenberg, and Muhammad Barham. Practical Solutions For Format-Preserving Encryption. *CoRR*, abs/1506.04113, 2015.

[14] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.